

REMARKS/ARGUMENTS

Claims 9 and 28 have been amended to clarify the antecedent basis for “file server” and “virus checker program”.

On page 2 of the Official Action, claims 1-7, 9, 12-15, 18, 20-26, 28, 31, and 38-39 were rejected under 35 U.S.C. 102(b) as being anticipated by Chen et al. (US 5,960,170). The applicant respectfully traverses.

The invention of applicant’s claim 1 is a method of using a virus checker in one file server to check for viruses in another file server. A data processing system includes at least one client, a first file server coupled to the client for data access of the client to at least one file in the first file server, and at least a second file server coupled to the first file server for data access of the second file server to the file in the first file server. The second file server is programmed with a virus checker program. The virus checker program is executable by the second file server to perform an anti-virus scan upon file data in random access memory of the second file server. The method includes the first file server responding to a request for access from the client to the file in the first file server by determining that an anti-virus scan of the file should be performed, and initiating the anti-virus scan of the file by sending to the second file server a request for the anti-virus scan including a specification of the file. Then the second file server responds to the request for the anti-virus scan by invoking the virus checker program to perform an anti-virus scan of the specified file by obtaining file data of the file from the first file server and storing the file data of the file into the random access memory of the second file server and performing the anti-virus scan upon the file data of the file in the random access memory.

Chen et al. (FIG. 1) shows a data processing system including at least one client (300a, 300b, 300c), a LAN server 350, and a virus detection server 400. Pursuant to a request for a virus scan, a virus detection object is produced by the virus detection server and is transmitted to a client for execution. The client receives and executes the virus detection object, and the results are transmitted to the virus detection server. The virus detection server uses the results to produce an additional virus detection object which is also transmitted to the client and executed so that the results can be transmitted to the virus detection server. The iterative production and execution of virus detection objects is continued until a determination is made as to whether the targeted file or data includes a virus. Upon a determination that a targeted file or data includes a virus, a vaccine specifically tailored to the conditions presented at the client and the type of virus detected is produced, preferably in the form of a virus treatment object. The request for a virus scan can be directly made or indirectly by a triggering event. (Abstract.) The request can be a programmed request from the client that does not require ongoing user initiation such that the scan is initiated without a request that is apparent to the user. (Col. 2, lines 3-7.) With reference to FIG. 2, in an initial step 205, a request for a virus scan is received, typically from a source external to the virus detection server 400 such as a client to be scanned. After receipt of the request, in step 210 it is determined by the virus detection server 400 whether a scan is to be performed. Preferably, a validation of the virus scan request is performed pursuant to the determination of whether a scan is to be performed. (Col. 6, lines 34-40.) FIG. 4A is a block diagram illustrating an embodiment of a virus detection server. (Col. 5, lines 1-2.) With

reference to FIG. 4A, the memory of the virus detection server is preferably configured to include routines for the iterative detection of viruses. The configurations are described in further detail with reference to the iterative virus detection module 450b of FIG. 4B. (Col. 10, lines 18-30.) Referring now to FIG. 4B, an embodiment of an iterative virus detection module ("IVDM") 450b is shown to include a scanning module 454, a virus pattern module 456, a virus rules module 458, a cleaning module 460, a cleaning pattern module 462, an access managing module 464, and an access data module 466. The iterative virus detection module 450b, and its referenced modules, includes routines for receiving virus detection requests, validating requests, producing virus detection and treatment objects, receiving the results of the execution of the virus detection and treatment objects, and using the results to produce additional virus detection and treatment objects to ultimately detect viruses and treat them. The iterative virus detection module 450b is typically implemented in software, but can also be implemented in hardware or firmware. (Col. 10, lines 52-67.)

“For a prior art reference to anticipate in terms of 35 U.S.C. § 102, every element of the claimed invention must be identically shown in a single reference.” Diversitech Corp. v. Century Steps, Inc., 7 U.S.P.Q.2d 1315, 1317 (Fed. Cir. 1988), quoted in In re Bond, 15 U.S.P.Q.2d 1566, 1567 (Fed. Cir. 1990) (vacating and remanding Board holding of anticipation; the elements must be arranged in the reference as in the claim under review, although this is not an *ipsis verbis* test).

It is respectfully submitted that Chen et al. does not anticipate the invention of applicant's claim 1 because the client and servers in Chen et al. do not operate in the fashion specified in applicant's claim 1.

In Chen et al., the virus detection server 400 responds to a triggering event indicating a need for a virus scan of a file in a client computer by sending virus scanning programs to the client, and the client executes these virus scanning programs to scan its file.

In contrast, in the method of applicant's claim 1, a first file server containing a file responds to a particular triggering event for initiating an anti-virus scan of the file (the first file server responding to a request for access from the client to the file in the first file server by determining that an anti-virus scan of the file should be performed). The first file server initiates the anti-virus scan of the file by sending to the second file server a request for the anti-virus scan including a specification of the file. Then the second file server responds to the request for the anti-virus scan by invoking the virus checker program to perform an anti-virus scan of the specified file by obtaining file data of the file from the first file server and storing the file data of the file into the random access memory of the second file server and performing the anti-virus scan upon the file data of the file in the random access memory.

In short, in Chen et al., when a need arises to check a file in a client computer, the virus checking server sends a virus checking program to the client computer, and the client computer executes the virus checking program to check the file in the client computer. In contrast, in the method of applicant's claim 1, when a need arises to check a file in a first file server, the first file

server sends the file data to the virus checking server, and the virus checking server executes the virus checking program to scan the file data in the virus checking server.

With respect to claims 2, 21 and 39, the cited portion of Chen et al. discloses that a client may request a virus scan and a virus detection server may validate a virus scan request to determine whether a virus scan should be performed. It is respectfully submitted, however, that Chen et al. col. 6, lines 34-48 fails to disclose that a file server containing a file determines that an anti-virus scan of the file should be performed when the client requests the file server to open the file and the file server finds that the file has not been checked for viruses.

With respect to claims 3, 22, and 39, it is not seen where Chen et al. discloses that a file server determines that the anti-virus scan of the file should be performed when the client requests a file to be closed after the client writes to the file.

With respect to claims 5 and 24, it is not seen where Chen et al. discloses a file server blocks clients from accessing the file from the time that the file server determines that the anti-virus scan of the file should be performed until the anti-virus scan is completed and fails to find a virus in the file.

With respect to claims 6 and 25, it is not seen where Chen et al. discloses that a file server determines that an additional anti-virus scan of the file should not be performed in response to the access of the file by the virus checker program.

With respect to claim 7, the cited portion of Chen et al. relates to the indexing of virus signatures, platform, virus type, virus identification, and virus information and criteria for determining which scanning and treatment routines to use. In contrast, the applicant's claim 7

relates to a file server maintaining in nonvolatile memory an indication of files that have not been checked for viruses, an indication of files that are in the process of being checked, and an indication of files that have been found to contain viruses. The state of a file with respect to whether the file has been or is being scanned for viruses should not be confused with criteria for determining which scanning and treatment routines to use.

With respect to claim 9, the cited portion of Chen et al. relates to the initiation of a triggering event outside of the virus checking server. In contrast, claim 9 relates to how a triggering event received by a file server is directed inside the file server to invoke a virus checker program in the file server to perform an anti-virus scan of a file. In particular, as defined in claim 9, the second file server reports a file access event to an operating system of the second file server, and the operating system of the second file server responds by invoking the virus checker program to perform the anti-virus scan of the file.

With respect to claim 12, the cited portion of Chen et al. relates to a group of computers that a user might seek to manage sharing a virus checking server. In contrast, the applicant's claim 12 relates to a first file server performing a load balancing procedure to select at least one of a second file server (programmed with a virus checker program) or a third file server (also programmed with a virus checker program) to perform an anti-virus scan of a file when the first file server determines that an anti-virus scan of the file should be performed.

Claim 13 is distinguished from Chen et al. as stated above for claims 1, 2, 3, and 12.

Claims 14 and 18 are distinguished from Chen et al. as stated above for claims 2 and 3.

Claims 15 and 28 are distinguished from Chen et al. as stated above for claim 9.

Claim 20 is distinguished from Chen et al. as stated above for claim 1.

Claim 26 is distinguished from Chen et al. as stated above for claim 7.

Claim 31 is distinguished from Chen et al. as stated above for claim 12.

Claim 38 is distinguished from Chen et al. as stated above for claim 13.

In paragraph 4 on page 7 of the Official Action, claim 10 was rejected under 35 U.S.C. 103(a) as being unpatentable over Chen et al., and further in view of Cassagnol et al (U.S. 6,385,727 B1.) Applicant respectfully traverses. The subject matter of the independent base claim 1 would not have been obvious from Chen et al. in view of the differences discussed above between Chen et al. and the subject matter of claim 1, and there is nothing in Cassagnol et al. that makes up for the lack of disclosure in Chen et al. with respect to the differences discussed above. In particular, the basic operation in the applicant's system (when a need arises to check a file in a first file server, the first file server sends the file data to the virus checking server, and the virus checking server executes the virus checking program to scan the file data in the virus checking server) is entirely opposite to the basic operation in Chen et al.'s system (when a need arises to check a file in a client computer, the virus checking server sends a virus checking program to the client computer, and the client computer executes the virus checking program to check the file in the client computer). In addition, the subject matter added by the express language in the dependent claim 10 does not result from the proposed combination of Chen et al. and Cassagnol et al.

Cassagnol et al. (col. 3, lines 25-30) teaches that a processor may have a kernel mode of operation and a user mode of operation, in which non-secure software may be executed in the

user mode and secure software may be executed in the kernel mode. However, this teaching is too general to suggest the specific construction defined in applicant's claim 10. In particular, it is not seen how the proposed combination of Chen et al. and Cassagnol et al. would provide the applicant's server for virus checking executing in the user mode that receives the request for the anti-virus scan from the first file server and forwards the request to a virus checker initiator driver executing in the kernel mode, and the virus checker initiator driver executing in the kernel mode initiates a file access event, and the virus checker program initiates the anti-virus scan of the file in response to the virus checker initiator driver initiating the file access event. (See applicant's FIG. 3 and applicant's specification, page 19 line 4 to page 20 line 10.)

Where the prior art references fail to teach a claim limitation, there must be "concrete evidence" in the record to support an obviousness rejection. "Basic knowledge" or "common sense" is insufficient. In re Zurko, 258 F.3d 1379, 1385-86, 59 U.S.P.Q.2d 1693, 1697 (Fed. Cir. 2001); In re Gordon et al., 733 F.2d 900, 902, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984) (mere fact that prior art could be modified by turning apparatus upside down does not make modification obvious unless prior art suggests desirability of modification); Ex Parte Kaiser, 194 U.S.P.Q. 47, 48 (PTO Bd. of Appeals 1975) (Examiner's failure to indicate anywhere in the record his reason for finding alteration of reference to be obvious militates against rejection).

In paragraph 5 on page 8 of the Official Action, claims 8, 11, 16, 17, 19, 27, 29-30, 32-37, and 40 were rejected under 35 U.S.C. 103(a) as being unpatentable over Chen et al., and further in view of Cassagnol et al., Lam et al. (US 6,385,727 B1), and Tzelnic et al. (US

5,948,062). Applicant respectfully traverses. The subject matter of the independent base claims would not have been obvious from Chen et al. in view of the differences discussed above between Chen et al. and the subject matter of the independent base claims, and there is nothing in Cassagnol et al., Lam et al., and Tzelnic et al. that makes up for the lack of disclosure in Chen with respect to the differences discussed above. In particular, the basic operation in the applicant's system (when a need arises to check a file in a first file server, the first file server sends the file data to the virus checking server, and the virus checking server executes the virus checking program to scan the file data in the virus checking server) is entirely opposite to the basic operation in Chen et al.'s system (when a need arises to check a file in a client computer, the virus checking server sends a virus checking program to the client computer, and the client computer executes the virus checking program to check the file in the client computer). Where the prior art references fail to teach a claim limitation, there must be "concrete evidence" in the record to support an obviousness rejection. "Basic knowledge" or "common sense" is insufficient. In re Zurko, 258 F.3d 1379, 1385-86, 59 U.S.P.Q.2d 1693, 1697 (Fed. Cir. 2001); In re Gordon et al., 733 F.2d 900, 902, 221 U.S.P.Q. 1125, 1127 (Fed. Cir. 1984) (mere fact that prior art could be modified by turning apparatus upside down does not make modification obvious unless prior art suggests desirability of modification); Ex Parte Kaiser, 194 U.S.P.Q. 47, 48 (PTO Bd. of Appeals 1975) (Examiner's failure to indicate anywhere in the record his reason for finding alteration of reference to be obvious militates against rejection).

With respect to applicant's claim 11, it is not seen where the memory 414 in FIG. 4A of Chen includes an "input/output manager in the operating system". The routines for the iterative

detection of viruses would not be in the operating system. In addition, it is not understood how the server component management API of Lam et al., which rejects a remote procedure call for an incompatible version, would suggest the applicant's "input/output manager in the operating system of the second file server receives a file access call from the virus checker initiator driver, and responds by directing a report of the file access event to the virus checker program." The applicant's virus checker initiator driver and the applicant's input/output manager are directed to solving the problem of how to stimulate the operating system of the NT file server to cause the conventional virus checker program to check an external file. Therefore, if the conventional virus checker program were upgraded in a fashion compatible with the Windows NT/2000 operating system, the upgraded virus checker program would continue to be invoked by the RPC client for virus checking in the network file server. (See applicant's FIG. 3 and applicant's specification, page 19 line 4 to page 20 line 10.) The cited portion of Lam et al. would not have suggested a solution to this problem because Lam et al. deals with rejecting a remote procedure call for an incompatible version.

With respect to claim 16, it is respectfully submitted that Chen et al. is distinguished as stated above with respect to claim 1 to the extent that claim 16 calls for "a virus checker program executing in the second server in the user mode, and the virus checker program responds by obtaining file data from the file in the first server and storing the file data in random access memory in the second server, and performing an anti-virus scan upon the file data in the random access memory in the second server." Chen et al. is also distinguished as stated above with respect to claim 11 with reference to the recitation in claim 16 of the operating system of the

second server including an input/output manager executing in the kernel mode. In view of the statement in the Official Action that Chen does not explicitly mention (1) processes executing in a user mode and processes executing in a kernel mode, (2) the role of the input/output manager, it is respectfully submitted that the cited portions of Chen et al. (col. 10, lines 20-23 and 20-23) fail to disclose “the server for virus checking forwards the request to a virus checker initiator driver executing in the second server in the kernel mode, and the virus checker initiator driver responds to receipt of the request by sending a file access call to the input/output manager.” The Official action cites Cassagnol and Lam for these features. Cassagnol (col. 3, lines 25-30) teaches that a processor may have a kernel mode of operation and a user mode of operation, in which non-secure software may be executed in the user mode and secure software may be executed in the kernel mode. However, this teaching is too general to suggest the specific construction defined in applicant’s claim 16. Lam et al. is distinguished as stated above with reference to claim 11. It is not understood how the server component management API of Lam et al., which rejects a remote procedure call for an incompatible version, would have suggested both the applicant’s virus checker initiator driver and the applicant’s input output manager, and would have also suggested the applicant’s virus checker initiator driver executing in the second server in the kernel mode, and the virus checker initiator driver responds to receipt of the request by sending a file access call to the input/output manager. Thus, it is respectfully submitted that it would not have been obvious to reconstruct the subject matter of applicant’s claim 16 by picking and choosing various things from Chen et al., Cassagnol et al., and Lam et al., adding the missing elements, and modifying that combination as proposed in the Official Action.

Hindsight reconstruction, using the applicant's specification itself as a guide, is improper because it fails to consider the subject matter of the invention "as a whole" and fails to consider the invention as of the date at which the invention was made. "[T]here must be some motivation, suggestion, or teaching of the desirability of making the specific combination that was made by the applicant." In re Lee, 277 F.3d 1338, 1343, 61 U.S.P.Q.2d 1430, 1435 (Fed. Cir. 2002) (quoting In re Dance, 160 F.3d 1339, 1343, 48 U.S.P.Q.2d 1635, 1637 (Fed. Cir. 1998)). "[T]eachings of references can be combined only if there is some suggestion or incentive to do so." In re Fine, 837 F.2d 1071, 1075, 5 U.S.P.Q.2d 1596, 1600 (Fed. Cir. 1988) (Emphasis in original) (quoting ACS Hosp. Sys., Inc. v. Montefiore Hosp., 732 F.2d 1572, 1577, 221 U.S.P.Q. 929, 933 (Fed. Cir. 1984)). "[P]articular findings must be made as to the reason the skilled artisan, with no knowledge of the claimed invention, would have selected these components for combination in the manner claimed." In re Kotzab, 217 F.3d 1365, 1371, 55 U.S.P.Q.2d 1313, 1317 (Fed. Cir. 2000). See, for example, Fromson v. Advance Offset Plate, Inc., 755 F.2d 1549, 1556, 225 U.S.P.Q. 26, 31 (Fed. Cir. 1985) (nothing of record plainly indicated that it would have been obvious to combine previously separate lithography steps into one process).

With respect to applicant's claim 19, see the applicant's remarks above with respect to claim 16.

With respect to applicant's claim 27, see the applicant's remarks above with respect to claim 8.

With respect to applicant's claim 29, see the applicant's remarks above with respect to claim 10.

With respect to applicant's claim 30, see the applicant's remarks above with respect to claim 11.

With respect to applicant's claims 34 and 37, see the applicant's remarks above with respect to claims 1, 16, and 20.

With respect to claim 40, see the applicant's remarks above with respect to claim 16.

With respect to claim 17, see the applicant's remarks above with respect to claim 1, 2, 3, and 12.

With respect to claim 32, see the applicant's remarks above with respect to claims 1, 2, 3, and 12.

With respect to claims 33 and 36, see the applicant's remarks above with respect to claims 2 and 3.

With respect to claim 35, see the applicant's remarks above with respect to claims 1, 2, 3, and 12.

In view of the above, reconsideration is respectfully requested, and early allowance is earnestly solicited.

Respectfully submitted,

Feb. 2, 2005



Richard C. Auchterlonie
Reg. No. 30,607

NOVAK DRUCE & QUIGG, LLP
1000 Louisiana, Suite 5320
Houston, TX 77002
713-751-0655